Lab 2: Process Management

East Tennessee State University

CSCI 4417/5417: Introduction to System Administration

Spring 2016

Jack Ramsey

# Purpose

Explore Windows and Ubuntu process management tools and concepts.

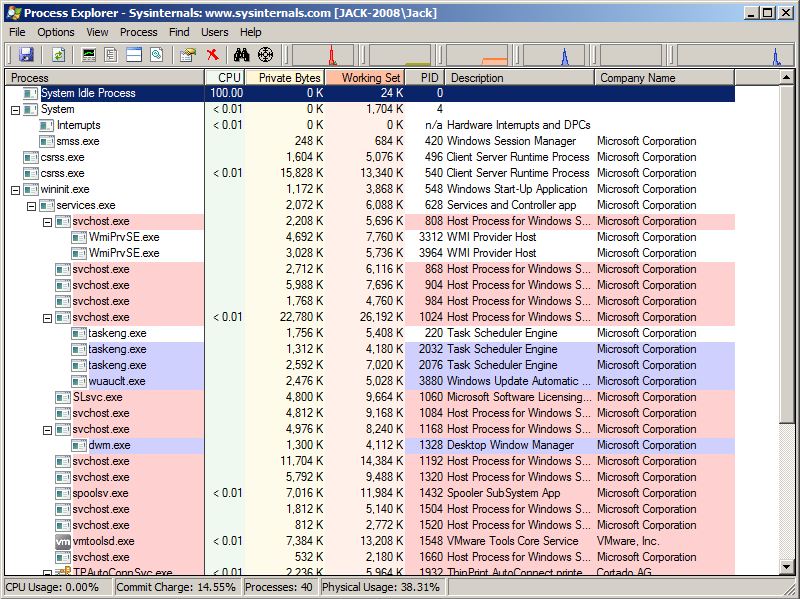
# Materials

* Windows Server 2012 R2 AWS Instance
* SysInternals Process Explorer v16.12
* Ubuntu 14.04 LTS AWS Instance

# Procedure and Results

## Windows

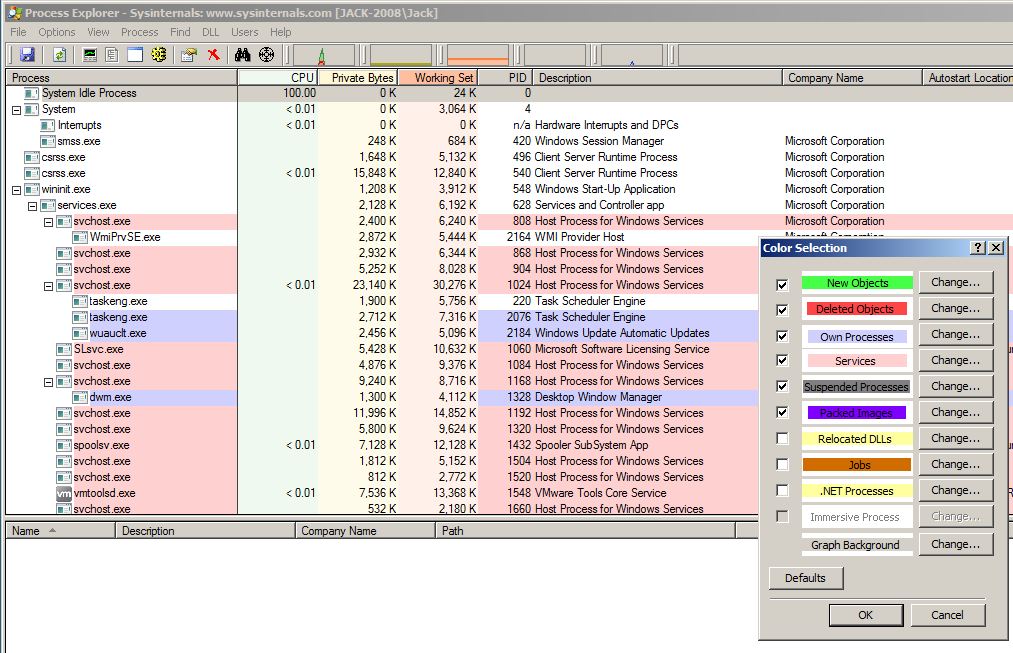
After launching the Windows 2008 VM, Process Explorer was downloaded from the Windows SysInternals website (<http://technet.microsoft.com/en-us/sysinternals/bb896653>). Process Explorer doesn’t require installation to run. It was launched in administrator mode by right-clicking *procexp.exe*, located in the Desktop\ProcessExplorer folder after downloading, and selecting “*Run as administrator*.” When Process Explorer was launched, a display of the system’s running processes in tree form was displayed (see Fig. 1).



*Figure 1: Initial display of Process Explorer*

### Process Grouping

Processes are displayed in Process Explorer in a tree format that illustrates the parent/child relationship between processes. Child processes are displayed below and indented from their parent processes. Furthermore, Process Explorer uses a highlighting scheme to further define process attributes. Clicking on **Options -> Configure Colors** displays the current color schema and allows the user to reconfigure it, if so desired (see Fig. 2).

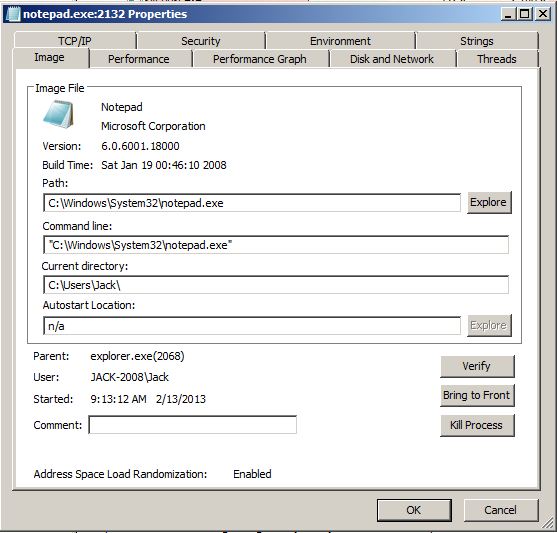


*Figure 2: Highlighting options in Process Explorer*

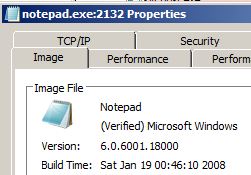
### Examining a Process’s Properties

Notepad (notepad.exe) was then launched in the VM. After locating the notepad.exe process in Process Explorer, its properties were displayed via double-clicking the process name (see Fig. 3). There are a number of tabs displayed in the properties window:

* Image: Provides meta-data about the process, including its name, producer, version, build time, path, command line, current directory, parent process, user, and start time. Process Explorer also checks for whether or not an image has been digitally signed by a certificate root authority trusted by the computer and displays the status of the check, which is either "Trusted" (signed), "Unsigned", or "Not verified” (signature has not been checked). Clicking the Verify button causes Process Explorer to check the signature of an image that has not been verified. The verification operation can result in Process Explorer contacting web sites to check for certificate validity (see Fig 4). Other options available in this tab include creating a comment, bringing the process to the front, and killing the process.

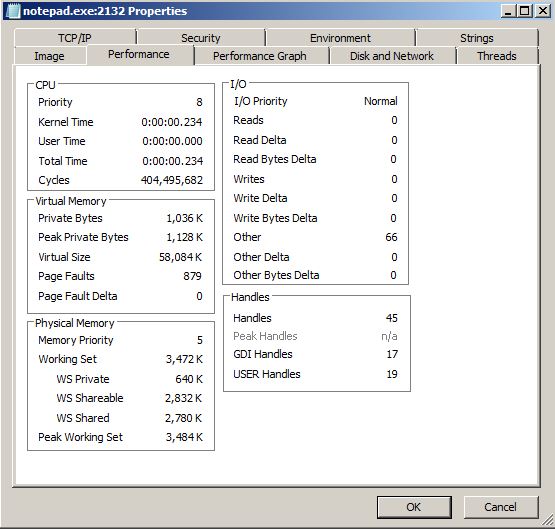


*Figure 3: Process Properties (notepad.exe)*

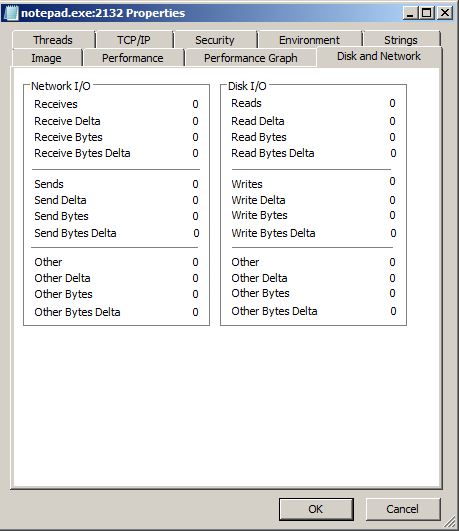


*Figure 4: Properties display after clicking "Verify" button (see Fig. 3)*

* The Performance tab displays a number of performance metrics, including CPU, Virtual Memory, Physical Memory, Input/Output (I/O), and Handles. Specific metrics associated with each category can be seen in Figure 5.
* The Performance Graph tab provides a graphical representation of CPU, Private Bytes, and I/O information.
* The Disk and Network tab provides information about the process’s input/output to disk and the network (see Fig. 6).

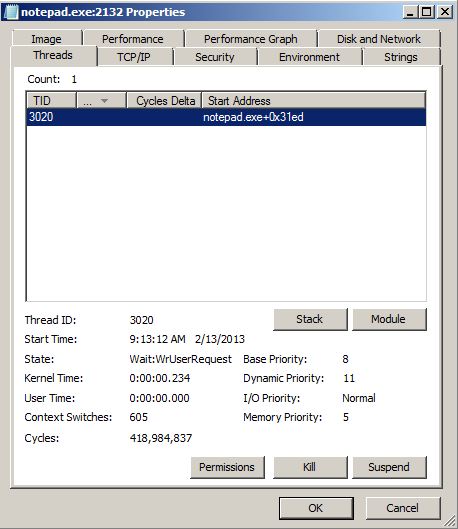


*Figure 5: Performance tab in Properties Applet*



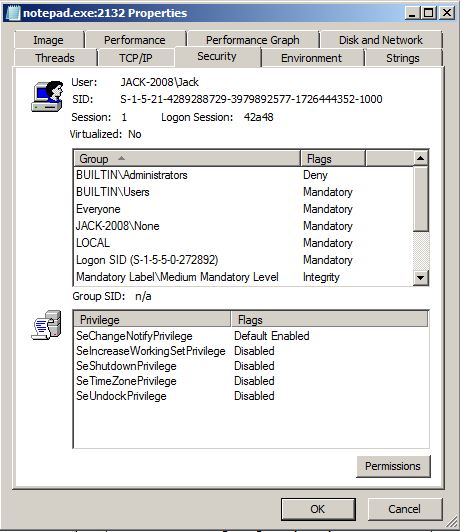
*Figure 6: Disk and Network information*

* The Threads tab displays the threads that are associated with the process, along with information specific to each thread (see Fig. 7), including Thread ID, Start Time, State, Kernel Time, User Time, Context Switches, Cycles, Base Priority, Dynamic Priority, I/O Priority, and Memory Priority. There are also options to display stack information, display module information, modify permissions, and kill or suspend the process.



*Figure 7: Threads tab on Process Properties applet*

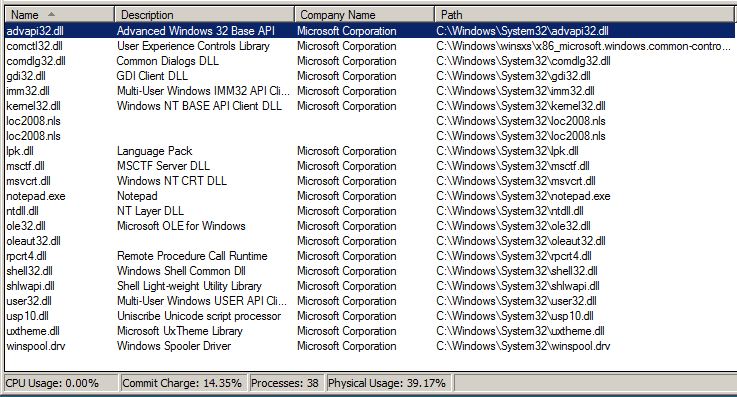
* The TCP/IP tab displays any active TCP and UDP endpoints owned by the process.
* The Security tab displays the list of groups and privileges listed in the security token of the process (see Fig. 8). The permissions button opens a permissions editor that shows the access permissions assigned to the process.
* The Environment tab displays environment variables associated with the process.
* The Strings tab displays all printable strings of at least 3 characters in length[[1]](#footnote-1).



*Figure 8: Security tab*

### Examining a Process’s DLLs

The bottom pane in Process Explorer can be set to display all of the application extensions (.dll) that are associated with the selected process (see Fig. 9).



*Figure 9: Bottom pane display of notepad.exe's .dlls*

Referring to Figure 9, it is evident that even a relatively unsophisticated application such as Notepad requires a number of supporting modules to function. For example, the application extension *gdi32.dll* contains functions for the Windows GDI (Graphical Device Interface) which assists windows in creating simple 2-dimensional objects (Uniblue Systems Limited, 2004). There aren't any 64-bit application extensions displayed because the installation of Windows Server 2008 in use for this lab is a 32-bit version.

 The target tool on the Process Explorer toolbar ( ) identifies a window’s process. Clicking on the icon, dragging it to an open window, and un-clicking will cause Process Explorer to highlight that window’s process in the Process Explorer.

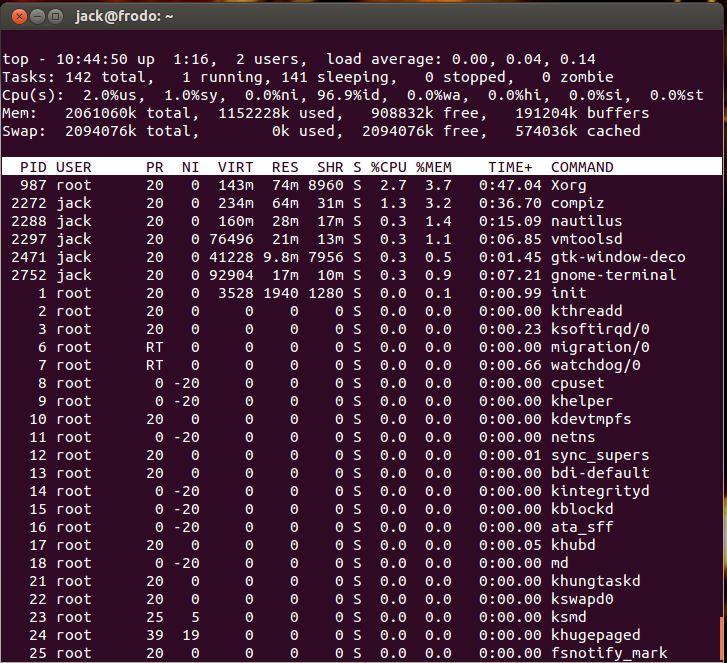
### Process Examination

The following processes were examined in accordance with lab instructions:

1. Notepad
   1. Name of process: *notepad.exe*
   2. What does it do? Simple text editor.
   3. What are its startup parameters? (Does it load at bootup or when a program is run?): Notepad loads when the program is run.
   4. What happens if we attempt to kill the process? It dies.
   5. If an application is associated with the process, what is it? Notepad is a child of explorer.exe
2. Explorer
   1. Name of process: *explorer.exe*
   2. What does it do? Parent process for user applications
   3. What are its startup parameters? (Does it load at bootup or when a program is run?): Explorer.exe loads at system startup.
   4. What happens if we attempt to kill the process: Several system functions died, such as the task bar and desktop icons. The system continued to operate without crashing, however. Functionality was limited to active processes only and it was necessary to restart the system to resume normal operations.
   5. If an application is associated with the process, what is it? Explorer.exe is also associated with the Windows Explorer application, used for file and folder management, etc.
3. Microsoft Distributed Transaction Coordinator
   1. Name of process: *msdtc.exe*
   2. What does it do? Manages transactions across multiple servers.
   3. What are its startup parameters? (Does it load at bootup or when a program is run?): msdtc.exe runs at system startup.
   4. What happens if we attempt to kill the process? It is re-spawned by its parent process (*TPAutoConnSvc.exe*).
   5. If an application is associated with the process, what is it? *TPAutoConnSvc.exe*  – Thin Print Auto Connect Service.
4. Windows Automatic Updates
   1. Name of process: *wuauclt.exe*
   2. What does it do? Manages Windows automatic updates
   3. What are its startup parameters? (Does it load at bootup or when a program is run?): Runs on startup, but can be configured not to by the user.
   4. What happens if we attempt to kill the process? It dies.
   5. If an application is associated with the process, what is it? It is associated with *svchost.exe*, because it is a Windows Service.
5. Windows Session Manager
   1. Name of process: *smss.exe*
   2. What does it do? Responsible for starting the user session.
   3. What are its startup parameters? (Does it load at bootup or when a program is run?): Windows Session Manager is a System process that runs on system startup.
   4. What happens if we attempt to kill the process? It dies. System will continue to function normally until user either logs off or attempts to switch users. System will then hang.
   5. If an application is associated with the process, what is it?
6. Services
   1. Name of process: *services.exe*
   2. What does it do? Manages the starting and stopping of services (manually and automatically)
   3. What are its startup parameters? (Does it load at bootup or when a program is run?): It starts at bootup.
   4. What happens if we attempt to kill the process? It kills Windows and forces a restart. In fact, this action caused VMWare to hang up as well. It was necessary to reboot the physical machine to get everything working again.
   5. If an application is associated with the process, what is it? This process is associated with the Windows Startup application (*wininit.exe*).
7. Desktop Window Manger
   1. Name of process: *dwm.exe*
   2. What does it do? Manages windows effects.
   3. What are its startup parameters? (Does it load at bootup or when a program is run?): It loads at bootup.
   4. What happens if we attempt to kill the process? No apparent effect in Windows 2008. In Windows Vista or Windows 7, windows effects would probably be lost.
   5. If an application is associated with the process, what is it? This process is associated with *svchost.exe* because it is a service.
8. Winlogon
   1. Name of process: *winlogon.exe*
   2. What does it do? Manages log on subsystem; loading user profiles, locking the system when a screensaver is running, and verifying the operating system's activation key.
   3. What are its startup parameters? (Does it load at bootup or when a program is run?): It loads a system bootup.
   4. What happens if we attempt to kill the process: Current user is immediately logged out. Appeared to be equivalent to selecting **Start -> Log Off**.
   5. If an application is associated with the process, what is it?

## Procedure: Ubuntu Processes

The Ubuntu machine was then run in VMWare Player. After the system booted up, a terminal window was launched and the *top* command was run (see Fig. 10).

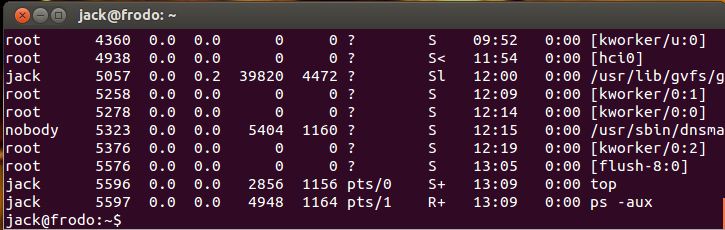


*Figure 10: Running the top command in Ubuntu*

The top five lines in this window provide information about the system up time, number of users, CPU load average, a breakdown of running tasks and their statuses, CPU usage, and memory usage. The table following the header information contains 12 columns:

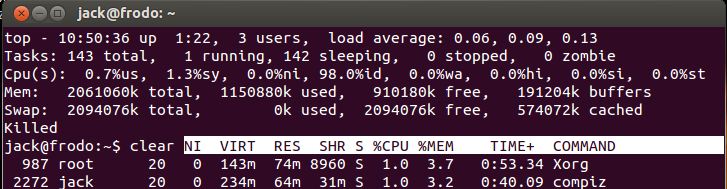
1. PID: The process id
2. User: The user who owns the process
3. PR: Process priority
4. NI: Nice value (a negative value indicates a higher priority)
5. VIRT: Virtual image; total amount of memory used by the task
6. RES: The resident size; non-swapped memory a process has used
7. SHR: The amount of shared memory used by a task/memory that could be potentially shared with other processes
8. S: Process status
9. %CPU: Percentage of CPU being utilized by the process
10. %MEM: Percentage of memory being utilized by the process
11. TIME+: Total CPU time the task has used since it started, in hundredths.
12. COMMAND: Command line or program name.

The command, *ps -aux* was then run, and the results examined (see Fig. 11).



*Figure 11: Partial display from ps -aux command in Ubuntu*

The last two lines (highlighted) show that the most recent processes launched are *top* and *ps -aux*. In the second terminal window, the command *kill -9 5596* was then run to kill the *top* process. Interestingly, the process was terminated, but the command prompt didn’t display under the last of the text, as is customary (see Fig. 12).



*Figure 12: Killed top process, but prompt didn't display prompt at the bottom*

# Observations

A modern computer operating system consists of a number of processes or applications that run “under the hood” to support and facilitate user, system, and hardware activities and interaction. Processes may be instantiated by the system, by a user, or by other processes. Familiarity with the concept of processes, their interaction, and their interdependence is a fundamental need for a systems administrator. Benefits that accrue from mastering this concept are a better understanding of the system’s architecture, greater ability to effectively troubleshoot problems, and the ability to fix problems on the fly rather than resorting to rebooting.

Knowledge and understanding of process names and functionality provide a greater understanding of the system architecture upon which the administrator’s system operates. A fundamental understanding of a system’s architecture enhances a sysadmin’s ability to maintain and upgrade his or her systems and to anticipate problems that might arise during maintenance. Though some environments may consist of homogenous systems, an understanding of multiple architectures, e.g., Windows and Linux, also enhances one’s ability to administer and maintain a more heterogeneous environment, which is common in modern distributed systems.

When problems do arise in a system, a sysadmin who knows how to examine running processes and troubleshoot problems will be more effective and will provide greater value to his or her employer. Applications such as Process Explorer for Windows and native Linux applications such as *top*, *ps*, and */proc* enable examination of the state of a system as it is running to isolate problems; ideally, on the fly. Successfully troubleshooting problems improves repair time and minimizes downtime, which is increasingly unacceptable to organizations that rely on information technology systems for daily operations. A keen grasp of process management can lead to an intuitive sense of how a system is functioning and what needs to be done when the system becomes unstable or fails entirely.

As noted before, rebooting a system is an inelegant way of repairing a problem. Without understanding the nature of the problem and taking positive steps toward its resolution, a simple system reboot will merely ensure that the problem will arise again. Thus, the time spent rebooting a system (during which time it is unavailable) leads to a greater likelihood that it will become unstable or fail again in the future, and increasing its overall downtime. Understanding how to examine running process, and therefore take more proactive or effective action in response to issues that arise, better enables a sysadmin to accomplish his or her goals and provide value to his or her organization in the form of effective operating systems with minimal downtime.

1. Information for these definitions came primarily from the Process Explorer help file. [↑](#footnote-ref-1)